COMMENTS

The enclosed is responsive to the Examiner's Office Action mailed January 16, 2002. At the time of the Examiner's mailing, claims 1 through 17 were pending. In response, the Applicant has: 1) canceled claims 6, 8, 15 and 17; 2) amended claims 1 through 5, 7, 9, 10 through 14, and 16; and, 3) added new claims 18 through 49. As such, claims 1 through 5, 7, 9 through 14, 16 and 18 through 49 are currently pending. The applicant respectfully requests reconsideration of the present application and the allowance of claims 1 through 5, 7, 9 through 14, 16 and 18 through 49.

The Examiner rejected claims 1, 7, 9, 10 and 16 under 35 USC 112, second paragraph, as being indefinite for reciting "first set of errors" and "determining if a threshold has been reached". In response the Applicant has amended claims 1, 7, 9, 10 and 16 so as to eliminate the phrase "first set of errors"; and has amended independent claims 1, 9 and 10 so as to provide clearer definition as to the term "threshold". As such, the Examiner's rejection has been overcome.

The Examiner rejected independent claims 1, 9 and 10 under 35 USC 103 as being unpatentable over US Patent No. 6,124,802 (hereinafter, "Ozaki") and US Patent No. 6,173,386 (hereinafter, "Key"). The Examiner also objected to claims 8 and 17 saying that both would be allowable if rewritten in independent form. As such, the Applicant has amended independent claims 1, 9 and 10 so as to recite subject matter related to that of claims 8 and 17. The Applicant thanks the Examiner for his objection to claims 8 and 17.

Believing independent claims 1, 9 and 10 to be patentable, the Applicant respectfully requests their allowance as well as the allowance of dependent claims 2 through 5, 7, 11 through 14 and 16. The applicant also respectfully requests the allowance of new claims 18 through 49.

If there are any additional charges, please charge them to our Deposit Account Number 02-2666. If a telephone conference would facilitate the prosecution of this application, the Examiner is invited to contact Robert B. O'Rourke at (408) 720-8300.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 5/15, 2002

Robert B. O'Rourke Reg. No. 46,972

12400 Wilshire Blvd.
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300

AMENDMENTS SHOWING CHANGES

Please cancel claims 6, 8, 15 and 17.

Please amend claims 1 through 5, 7, 9 through 15 and 16 as indicated.

1. (once amended) A method of managing a network switch having a processor card including a memory and a processing unit in the processor card comprising:

detecting an error;

determining [a type of]if the error is ignorable;

determining whether a threshold has been reached if the error is determined to be ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

performing a hitless rebuild in the processor card [when the type of the error is in a first set of errors and] if the threshold has not been reached.

2. (once amended) The method of claim 1, wherein the [step of] performing of the hitless rebuild includes:

performing an initialization of the memory; and,

protecting a portion of the memory from access by the processing unit during the initialization.

3. (once amended) The method of claim 1, wherein the performing of the [a]hitless rebuild further comprises [includes]protecting a portion of the memory that contains a set of routing tables.

4. (once amended) The method of claim 1, wherein the performing of the [a]hitless rebuild further comprises[includes] protecting a portion of the memory that contains a set of state tables.

5. (once amended) The method of claim 1, wherein the memory is accessed through a set of memory addresses and the performing of a hitless rebuild [includes]further comprises preventing the processing unit from accessing a predetermined set of memory addresses in the set of memory addresses.

2. (canceled) [The method of claim 1, wherein determining the type of the error includes identifying a non-ignorable error.]

3. (once amended) The method of claim 1, further [including]comprising setting the processing unit to enter into a degraded mode [when the type of the error is in the first set of errors]if the error is not ignorable and if the threshold has been reached.

4. (canceled) [The method of claim 1, wherein determining whether the threshold has been reached includes:

determining a number of hitless rebuilds that has been performed in the processor card;

determining a time period from a first hitless rebuild time and a second hitless rebuild time; and,

determining whether the number of hitless rebuilds versus the time period has reached a ratio.]

5. (once amended) An apparatus for managing a network switch having a processor card including a memory and a processing unit in the processor card comprising:

means for detecting an error;

means for determining [a type of]if the error is ignorable;

means for determining whether a threshold has been reached if the error is determined to be ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

means for performing a hitless rebuild in the processor card [when the type of the error is in a first set of errors and]if the threshold has not been reached.(once amended) An article comprising a [computer]machine readable medium having instructions stored thereon[,] which, when executed, cause[s] a method to be performed, the method comprising:

[detection of]detecting an error;

[determination of a type of]determining if the error is an ignorable error;

[determination of]determining whether a threshold has been reached if the error is determined to be an ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

[performance of]performing a hitless rebuild in a processor card [when the type of the error is in a first set of errors and]if the threshold has not been reached.

6. (once amended) The article of claim 10, wherein the [computer readable medium further having instructions stored thereon, which when executed, causes]the method further comprises:

[performance of]performing an initialization of the memory; and,

[protection of]protecting a portion of the memory from access by the processing unit during the initialization.

7. (once amended) The article of claim 10, wherein [the computer readable medium further having instructions stored thereon, which when executed, causes]the method further comprises:

[performance of]the performing of a hitless rebuild further includes [the step of] protecting a portion of the memory that contains a set of routing tables.

8. (once amended) The article of claim 10, wherein the [computer readable medium further having instructions stored thereon, which when executed, causes]method further comprises:

protecting a portion of the memory that contains a set of state tables.

9. (once amended) The article of claim 10, wherein the [memory is accessed through a set of memory addresses and the computer readable medium further having instructions stored thereon, which when executed, causes]memory further comprises:

[prevention of]preventing a processing unit from accessing a predetermined set of memory addresses in the set of memory addresses.

10. (canceled) [The article of claim 10, wherein the computer readable medium further having instructions stored thereon, which when executed, causes:

identification of a non-ignorable error].

11. (once amended) The article of claim 10, wherein the [computer readable medium further having instructions stored thereon, which when executed, causes]method further comprises:

setting of the processing unit to enter into a degraded mode [when the type of the error is in the first set of errors and]if the threshold has been reached.:

12. (canceled) [The article of claim 10, wherein the computer readable medium further having instructions stored thereon, which when executed, causes:

determination of a number of hitless rebuilds that has been performed in the processor card;

determination of a time period from a first hitless rebuild time and a second hitless rebuild time; and,

determination whether the number of hitless rebuilds versus the time period has reached a ratio.]

Please add new claims 18 through 49.

18. (new) A method, comprising:

re-initializing software that is executed on a networking device card, said re-initializing in response to an error, said networking device card comprising a processor that executes said software, said card comprising a memory that stores information that can be used by said processor to execute said software, said information further comprising a routing table, said re-initializing not deleting said routing table information from said memory.

19. (new) The method of claim 18 wherein said memory further comprises non-volatile memory.

20. (new) The method of claim 18 wherein said memory further comprises volatile memory.

21. (new) The method of claim 20 wherein said volatile memory further comprises random access memory.

22. (new) The method of claim 21 wherein said random access memory further comprises dynamic random access memory.

23. (new) A method, comprising:

re-initializing software that is executed on a networking device card, said re-initializing in response to an error, said networking device card comprising a processor that executes said software, said card comprising a memory that stores information that can be used by said processor to execute said software, said information further comprising a state table, said re-initializing not deleting said state table information from said memory.

24. (new) The method of claim 23 wherein said memory further comprises non-volatile memory.

25. (new) The method of claim 23 wherein said memory further comprises volatile memory.

26. (new) The method of claim 25 wherein said volatile memory further comprises random access memory.

27. (new) The method of claim 26 wherein said random access memory further comprises dynamic random access memory.

28. (new) An apparatus, comprising:

a networking device card, said networking device card comprising a processor that executes re-initializable software, said card comprising a memory that stores information that can be used by said processor to execute said re-intializable software, said information further comprising a routing table, said re-initializable software re-initialized in response to an error, said re-initializing not deleting said routing table information from said memory.

29. (new) The method of claim 28 wherein said memory further comprises non-volatile memory.

30. (new) The method of claim 28 wherein said memory further comprises volatile memory.

31. (new) The method of claim 30 wherein said volatile memory further comprises random access memory.

32. (new) The method of claim 31 wherein said random access memory further comprises dynamic random access memory.

33. (new) An apparatus, comprising:

a networking device card, said networking device card comprising a processor that executes re-initializable software, said card comprising a memory that stores information that can be used by said processor to execute said re-intializable software, said information further comprising a state table, said re-initializable software re-initialized in response to an error, said re-initializing not deleting said state table information from said memory.

34. (new) The method of claim 33 wherein said memory further comprises non-volatile memory.

35. (new) The method of claim 33 wherein said memory further comprises volatile memory.

36. (new) The method of claim 35 wherein said volatile memory further comprises random access memory.

37. (new) The method of claim 36 wherein said random access memory further comprises dynamic random access memory.

38. (new) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

re-initializing software that is executed on a networking device card, said re-initializing in response to an error, said networking device card comprising a processor that executes said software, said card comprising a memory that stores information that can be used by said processor to execute said software, said information further comprising a routing table, said re-initializing not deleting said routing table information from said memory.

39. (new) The article of claim 38 wherein said memory further comprises non-volatile memory.

40. (new) The article of claim 38 wherein said article is non-volatile memory.

41. (new) The article of claim 38 wherein said memory further comprises volatile memory.

42. (new) The article of claim 41 wherein said volatile memory further comprises random access memory.

43. (new) The article of claim 42 wherein said random access memory further comprises dynamic random access memory.

44. (new) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

re-initializing software that is executed on a networking device card, said re-initializing in response to an error, said networking device card comprising a processor that executes said software, said card comprising a memory that stores information that can be used by said processor to execute said software, said information further comprising a state table, said re-initializing not deleting said state table information from said memory.

45. (new) The method of claim 44 wherein said memory further comprises non-volatile memory.

46. (new) The method of claim 44 wherein said article is non-volatile memory.

47. (new) The method of claim 44 wherein said memory further comprises volatile memory.

48. (new) The method of claim 47 wherein said volatile memory further comprises random access memory.

49. (new) The method of claim 48 wherein said random access memory further comprises dynamic random access memory.